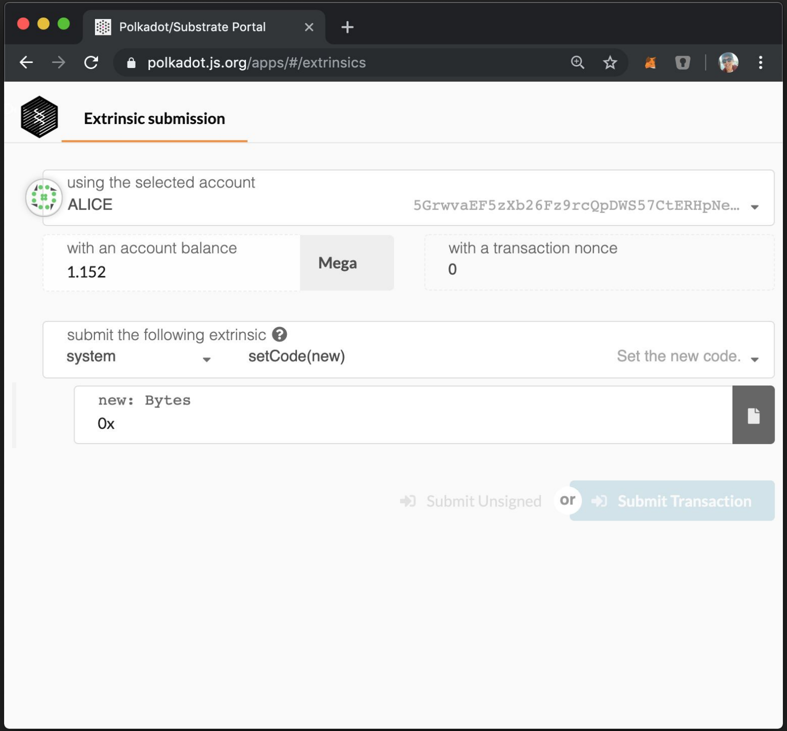
Substrate Node Template

Maggie Dong
Software developer @ Parity Technologies Ltd.

maggiedong@parity.io | @maggiedong

Polkadot JS Apps

- Generalized and hosted UI
- Quickly test new functionality
- Loaded with general tools like:
  - Creating transactions
  - Read storage
  - See events
  - and way more…
- Great for development


Polkadot/Substrate Portal

polkadot.js.org/apps/#/extrinsics

Extrinsic submission

using the selected account
ALICE 5GrwvaEF5zXb26Fz9rcQpDWS57CtERHpNe…

with an account balance
1.152 Mega

with a transaction nonce
0

submit the following extrinsic
system setCode(new) Set the new code.

new: Bytes
0x

Submit Unsigned or Submit Transaction

parity

模块定义概览

```
use support::{decl_module, decl_storage, decl_event,...};
pub trait Trait: system::Trait {...}

decl_storage! {...}
decl_event! {...}
decl_error! {...}
decl_module! {...}
impl<T: Trait> Module<T> {...}
```

引入和定义关联类型

```
pub trait Trait: system::Trait {
    type Event: From<Event<Self>> + Into<<Self as system::Trait>::Event>;
}
```

...and it inherits from `system::Trait`:

```
// From `system` pallet
pub trait Trait: 'static + Eq + Clone {
    type Origin: …
     type Call: …
     type Index: …
    type BlockNumber: ...
```

parity

定义存储

```
decl_storage! {
    trait Store for Module<T: Trait> as TemplateModule {
        // Here we are declaring a StorageValue, `Something` as an Option<u32>
        // `get(fn something)` defines a getter function
        // Getter called with `Self::thing()`
        Something get(fn something): Option<u32>;
        // Here we are declaring a StorageMap `SomeMap` from an AccountId to
        // a Hash.
        // Getter called with `Self::some_map(account_id)`
        SomeMap get(fn some_map): map hasher(identity) T::AccountId => u32;
    }
}
```

parity

定义事件

```
decl_event!(
    pub enum Event<T> where AccountId = <T as system::Trait>::AccountId {
        /// Event `SomethingStored` is declared with a parameter of the type `u32`
and `AccountId`
        SomethingStored(u32, AccountId),
    }
);
```

定义可调用函数

```
decl_module! {
  pub struct Module<T: Trait> for enum Call where origin: T::Origin {
    fn deposit_event<T>() = default; // The default deposit_event definition

    pub fn do_something(origin, something: u32) -> Result {
      let sender = ensure_signed(origin)?; // Check for transaction
      <Something::put(something); // Put a value into a StorageValue
      Self::deposit_event(RawEvent::SomethingStored(something, who)); // Emit Event
      Ok(()) // Return Ok at the end of a function
}}}
```

parity

定义公共和私有函数

```
impl<T: Trait> Module<T> {
  fn mint(to: T::AccountId, id: T::Hash) -> Result {...}
  fn transfer(from: T::AccountId, to: T::AccountId, id: T::Hash) -> Result {
      ...
  }
}
```

如果定义为 `pub`，其它模块也可以调用。

Coding

https://github.com/SubstrateCourse/substrate-node-template

parity

帮助链接

- https://www.substrate.io/
- https://www.substrate.io/tutorials
- Substrate Node Template, Front-end Template
- Official: Substrate Github, Polkadot Github
- 官方中文：知乎专栏，视频教学 (BiliBili)
- Rust: Rust programming book(中英文)

如何参与开源项目

- Star & watch & fork
- Check issues and PRs regularly
- Join riot/discord channels
- Feel free to ask any questions,
  stupid ones are always welcomed
- Follow the instructions to contribute

parity